Assignment 09: Data Scraping

Yared S. Asfaw

Total points:

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on data scraping.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_09_Data_Scraping.Rmd") prior to submission.

Set up

1. Set up your session:

- Check your working directory
- Load the packages `tidyverse`, `rvest`, and any others you end up using.
- Set your ggplot theme

```
# Checking the working directory
getwd()
```

```
## [1] "E:/EDA/Environmental_Data_Analytics_2022"
```

```
# Loading packages
library(tidyverse)
library(lubridate)
library(viridis)
library(rvest)
library(dataRetrieval)
library(tidycensus)

# Setting a theme
mytheme <- theme_classic() +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")
theme_set(mytheme)
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2020 Municipal Local Water Supply Plan (LWSP):

- Navigate to https://www.ncwater.org/WUDC/app/LWSP/search.php
- Change the date from 2020 to 2019 in the upper right corner.
- Scroll down and select the LWSP link next to Durham Municipality.
- Note the web address: https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2020

Indicate this website as the as the URL to be scraped. (In other words, read the contents into an `rvest` webpage object.)

```r
# Scraping the website
the_website <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2020')
the_website
```

```
## {html_document}
## <html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
## [1] <head>\n<title>DWR :: Local Water Supply Planning</title>\n<meta http-equ ...
## [2] <body id="plan">\r\n<!--<div id="division-header">\r\n<a name="top" href= ...
```

3. The data we want to collect are listed below:

- From the "1. System Information" section:

- Water system name

- PSWID

- Ownership

- From the "3. Water Supply Sources" section:

- Average Daily Use (MGD) - for each month

In the code chunk below scrape these values, assigning them to three separate variables.

> HINT: The first value should be "Durham", the second "03-32-010", the third "Municipality", and the last should be a vector of 12 numeric values, with the first value being 36.0100.

```r
# Scrapping variables of interests
water.system.name <- the_website %>%
  html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>%
  html_text()
pwsid <- the_website %>%
  html_nodes('td tr:nth-child(1) td:nth-child(5)') %>%
  html_text()
ownership <- the_website %>%
  html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>%
  html_text()
max.withdrawals.mgd <- the_website %>%
  html_nodes('th~ td+ td') %>%
  html_text()
```

4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

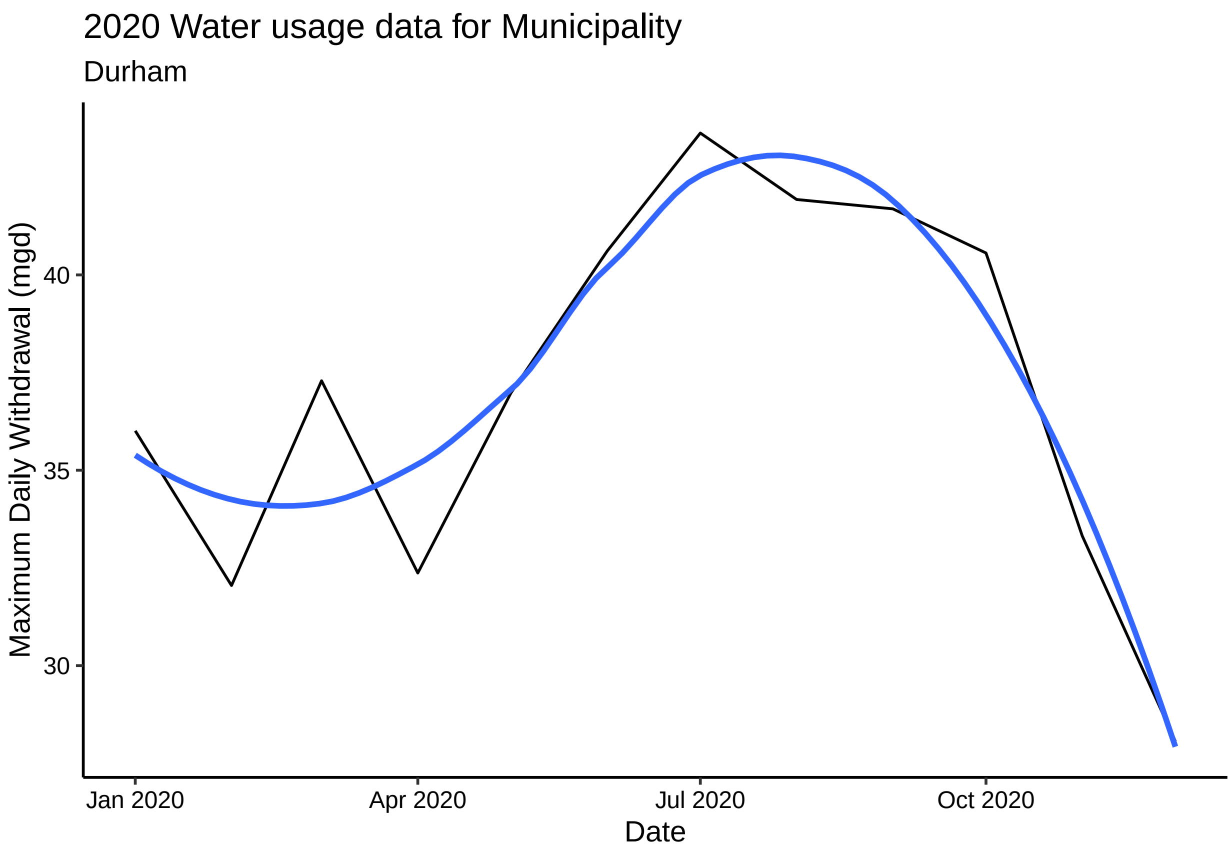
   TIP: Use `rep()` to repeat a value when creating a dataframe.

   NOTE: It's likely you won't be able to scrape the monthly widthrawal data in order. You can overcome this by creating a month column in the same order the data are scraped: Jan, May, Sept, Feb, etc…

```r
#Create a dataframe of withdrawals
months_in_the_year <- c('January', 'May', 'September', 'February', 'June','October',
                        'March', 'July', 'November', 'April', 'August', 'December')
df_withdrawals <- data.frame("Month" = months_in_the_year,
                  "Year" = rep(2020,12),
                   "Water_system_name" = water.system.name,
                  "PWSID"= pwsid,
                  "Ownership"= ownership,
              "Max_withdrawals_mgd"=as.numeric(max.withdrawals.mgd))  %>%
  mutate(Date = my(paste(Month,"-",Year))) %>%
  arrange(Date)
```

5. Plot the max daily withdrawals across the months for 2020

```r
# Plot
ggplot(df_withdrawals,aes(x=Date,y=Max_withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2020 Water usage data for",ownership),
       subtitle = water.system.name,
       y="Maximum Daily Withdrawal (mgd)",
       x="Date")
```



6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site scraped**.

```
# Constructing a function
scrape.it <- function(the_year, pwsid)
{
  if (the_year %in% c(1997, 2002, 2006:2021))
    {
  # Setting inputs for creating the function
  # Constructing the scraping web address, i.e. its URL
      the_base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid='
      the_scrape_url <- paste0(the_base_url, pwsid, '&', 'year=',the_year)
  # Retrieving the website contents
      the_website <- read_html(the_scrape_url)
  # Setting the elements address variables (determined under question # 3)
      water.system.name_tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'
      pswid_tag <- 'td tr:nth-child(1) td:nth-child(5)'
      ownership_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'
      max.withdrawals.mgd_tag <- 'th~ td+ td'

  #Scrapping the data items
      water.system.name <- the_website %>% html_nodes(water.system.name_tag) %>%
      html_text()
      pswid <- the_website %>%   html_nodes(pswid_tag) %>%  html_text()
      ownership <- the_website %>% html_nodes(ownership_tag) %>% html_text()
      max.withdrawals.mgd <- the_website %>% html_nodes(max.withdrawals.mgd_tag) %>%
      html_text()

  # Constructing a dataframe from the scraped data
  # Setting the months in the order they appear in the website
      months_in_the_year <- c('January', 'May', 'September', 'February', 'June',
                              'October', 'March', 'July', 'November', 'April',
      df_withdrawals <- data.frame("Month"= months_in_the_year,
                  "Year" = rep(the_year,12),
                  "Water_system_name" = water.system.name,
                  "PWSID"= pswid,
                  "Ownership"= ownership,
                  "Max_withdrawals_mgd" = as.numeric(max.withdrawals.mgd)) %>%
                   mutate(Date = my(paste(Month,"-",Year))) %>%
                   arrange(Date)

      return(df_withdrawals)
    }
  else
      return(paste("No data available for the year :", the_year))
 }
# Checking the function for the year not available on the website (eg. year 2000)
scrape.it(2000,'03-32-010')
```

```
## [1] "No data available for the year : 2000"
```

7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID='03-32-010') for each month in 2015

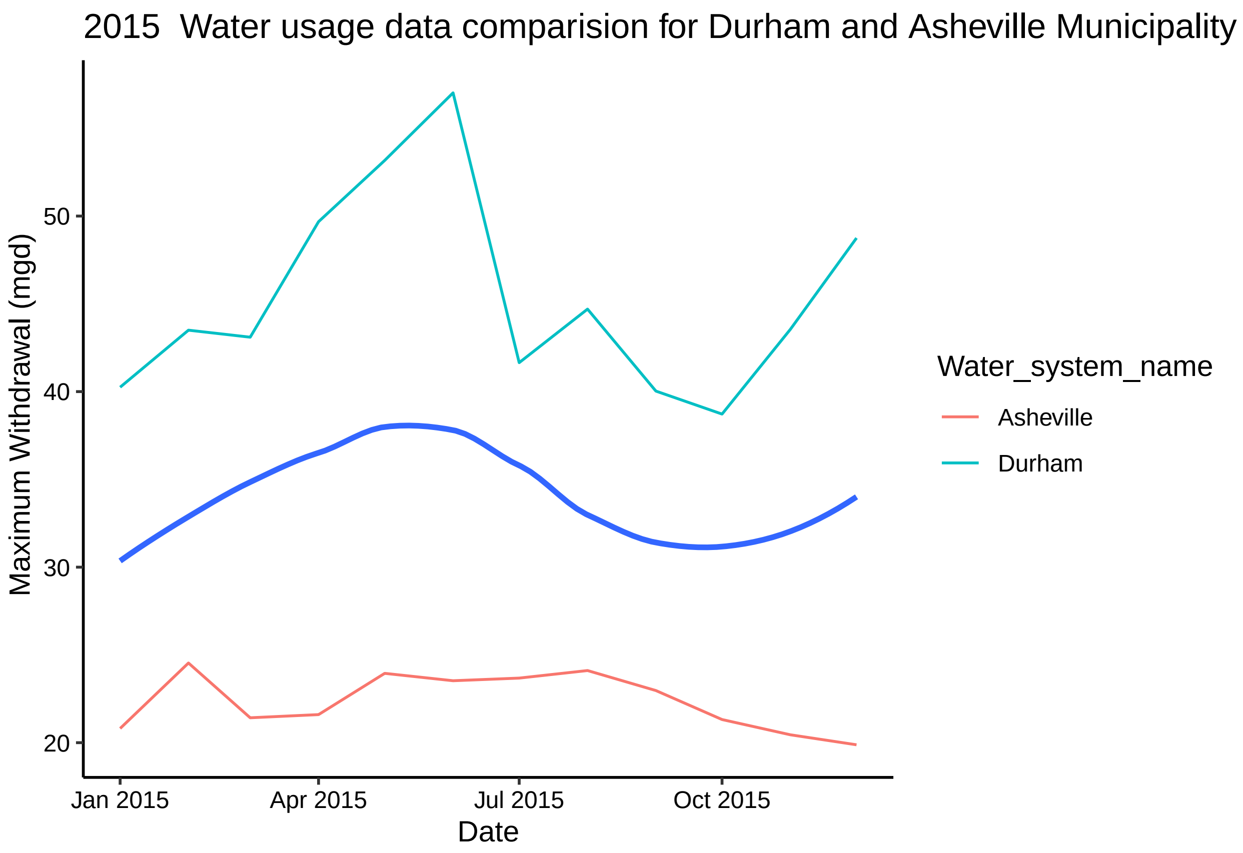
```
# Running the function to extract and plot max daily withdrawals for Durham for each month in 2015
Durham_df_2015 <- scrape.it(2015,'03-32-010')
view(Durham_df_2015)
```

8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares the Asheville to Durham's water withdrawals.

```
# Running the function to extract data for Asheville in 2015
Asheville_df_2015 <- scrape.it(2015,'01-11-010')
view(Asheville_df_2015)

# Combining the two dataframes and plotting comparision plots
combined_df <- rbind(Durham_df_2015, Asheville_df_2015)

ggplot(combined_df,aes(x=Date,y=Max_withdrawals_mgd)) +
  geom_line(aes(color = Water_system_name)) +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste(combined_df$Year, " Water usage data comparision for Durham and Asheville",
       ownership),
       y=" Maximum Withdrawal (mgd)",
       x="Date")
```

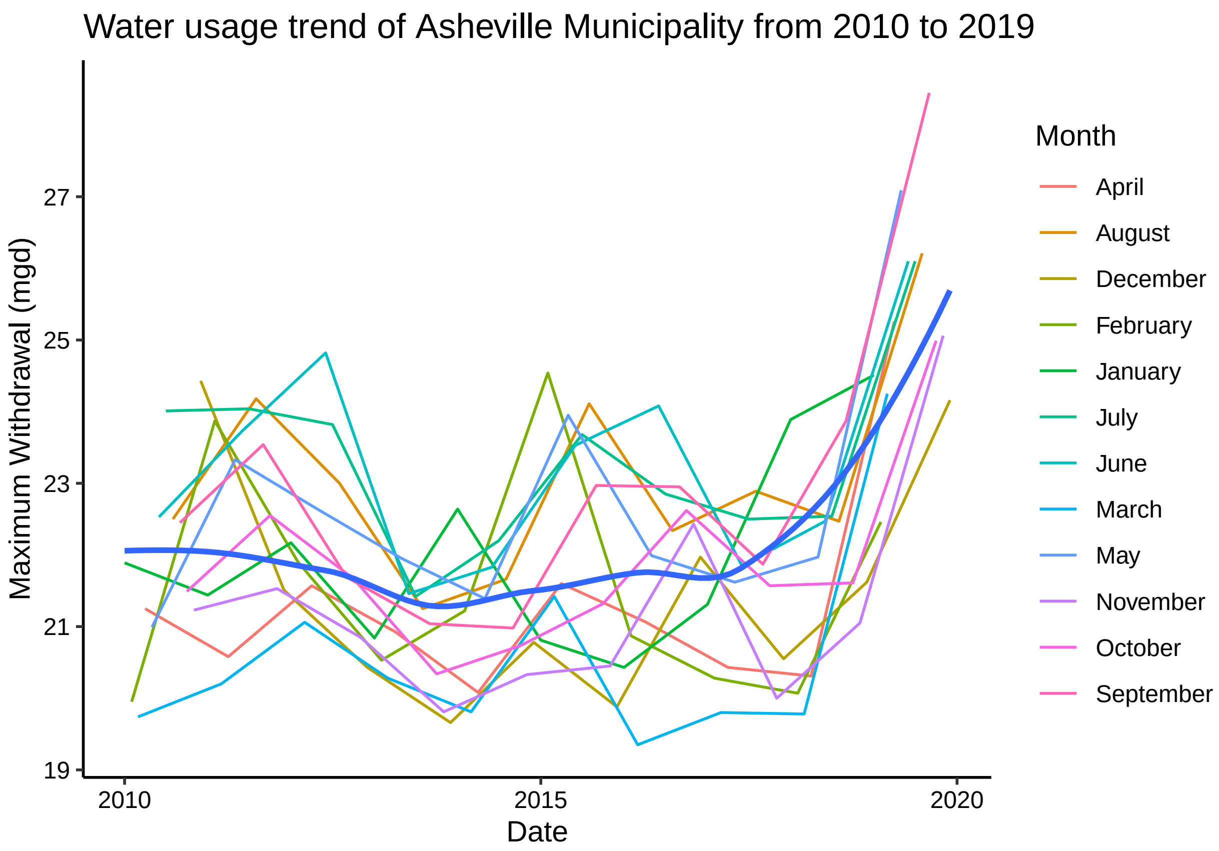


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2019.Add a smoothed line to the plot.

```
# Setting the inputs to scrape the years from 2010 to 2019 for Asheville
the_years = rep(2010:2019)
pwsid = '01-11-010'

# Using lapply to apply the scrape function
Asheville_df_max_DW <- lapply(X = the_years,
                  FUN = scrape.it,
                  pwsid=pwsid)
#Conflating the returned dataframes into a single dataframe
Asheville_df_max_DW <- bind_rows(Asheville_df_max_DW)

#Plotting the maximum daily withdrawals by months for each year
ggplot(Asheville_df_max_DW,aes(x=Date, y=Max_withdrawals_mgd)) +
  geom_line(aes(color=Month)) +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("Water usage trend of Asheville",ownership, "from 2010 to 2019"),
       subtitle = Asheville_df_max_DW$Water.system.name,
       y="Maximum Withdrawal (mgd)",
       x="Date")
```



Question: Just by looking at the plot (i.e. not running statistics), does Asheville have a trend in water usage over time?

The maximum water withdrawal of Asheville Municipality from 2010 to 2019 show a different trend over the periods. From 2010 to around 2014, it shows a slight decreasing trend, from around 2014 to around 2017, the maximum water withdrawal trend show a slight increasing trend from where it was in around 2014. However, after around 2017, the maximum water withdrawal show a dramatical increasing trend until 2019. In general, looking at the overall period (from 2010 to 2019), yes the water usage has a trend overtime, and this trend is slight decreasing at the begining, then a slight increasing and finally a dramatic increasing trends overtime.